

## Claims

What is claimed is:

- 1 1. In a network including a client system in communication with a server system  
2 hosting an application program written for execution as a single program unit and  
3 including software components, a method of executing the application program  
4 comprising:
  - 5 identifying one or more of the software components of the application  
6 program as a candidate for partitioning;
  - 7 generating at the server system a plurality of new software components  
8 corresponding to one of the identified software component candidates;
  - 9 generating a protocol to be used by the new software components to  
10 communicate with each other during an execution of the application program;  
11 and  
12 transmitting one of the plurality of new software components to the client  
13 system for execution at the client system, the new software component  
14 transmitted to the client system communicating with one of the other of the  
15 plurality of new software components at the server system using the generated  
16 protocol when the application program is executed.
- 1 2. The method of claim 1 further comprising wrapping the identified software  
2 component candidate in one of the new software components generated at the  
3 server system.

- 1 3. The method of claim 2 further comprising migrating the wrapped software
  - 2 component from one generated new software component to another generated
  - 3 new software component over the network.
  - 1 4. The method of claim 2 wherein the identified software component candidate is
  - 2 wrapped in the new software component that is transmitted to the client system
  - 3 for execution.
  - 1 5. The method of claim 4 wherein the new software components communicate with
  - 2 each other using a component protocol.
  - 1 6. The method of claim 2 wherein the identified software component is wrapped in a
  - 2 new software component at the server system for execution at the server system.
  - 1 7. The method of claim 6 wherein the new software components communicate with
  - 2 each other using an object protocol.
  - 1 8. The method of claim 1 further comprising including in the identified software
  - 2 component candidate a description that facilitates identification of that software
  - 3 component as a candidate for partitioning.
  - 1 9. The method of claim 1 wherein the step of identifying the software component as
  - 2 a candidate for partitioning includes determining that such software component
  - 3 has a software element that relates to a user interface.
  - 1 10. The method of claim 9 wherein the software element is an external interface of
  - 2 the identified software component candidate.

- 1 11. The method of claim 1 wherein the step of generating the plurality of software  
2 components corresponding to the identified software component candidate  
3 includes replicating external interfaces of the identified software component  
4 candidate for inclusion in each generated new software component.
- 1 12. The method of claim 1 wherein the steps of identifying one of the software  
2 components as a candidate for partitioning, generating the plurality of  
3 corresponding software components, and generating the protocol occur at run  
4 time of the application program.
- 1 13. The method of claim 1 further comprising analyzing the identified software  
2 component candidate to determine whether to execute the identified software  
3 component candidate at the server system rather than partition the identified  
4 software component candidate for execution at the client system.
- 1 14. In a network including a client system in communication with a server system, a  
2 method of partitioning a software component for dividing execution of the  
3 software component between the client and server systems, the method  
4 comprising:  
5       analyzing a first software component to determine whether the first  
6 software component is to be partitioned;  
7       if the first software component is to be partitioned:  
8           (a) dynamically generating a plurality of new software components  
9           corresponding to the first software component and a protocol to be used  
10          by the dynamically generated new software components for

communicating with each other, and (b) transmitting one of the dynamically generated new software components to the client system for execution at the client system and for communication with another of the generated new software components at the server system using the dynamically generated protocol;

otherwise, executing the first software component at the server and

communicating with the client system using a remote graphics protocol.

15. The method of claim 14 further comprising determining, if the first software component is to be partitioned, whether the first software component is to execute on the client system.
  16. The method of claim 15 wherein the dynamically generated protocol is a component protocol when the first software component is to execute at the client system.
  17. The method of claim 15 wherein the dynamically generated protocol is an object protocol when the first software component is to execute at the server system.
  18. The method of claim 15 further comprising providing an integrated development environment in which to analyze the first software component to determine whether the first software component is to be partitioned and to partition the first software component if the first software component is to be partitioned.
  19. In a network including a client system in communication with a server system, a method of executing an application program comprised of a user-interface

3 software component and a non-user-interface software component, the method  
4 comprising:

5 (a) generating a plurality of new software components corresponding to the  
6 user-interface software component of the application program;

7 (b) wrapping the user-interface software component with one of the new  
8 software components;

9 (c) transmitting one of the new software components to the client system;

10 (d) communicating with the new software component at the client system  
11 using a dynamically generated protocol when the user-interface software  
12 component is executed; and

13 (e) communicating with the client system using a remote graphics protocol  
14 when the non-user-interface software component is executed.

1 20. The method of claim 19 further comprising determining whether to execute the  
2 user-interface software component at the client system, and wherein the  
3 dynamically generated protocol is a component protocol if the user-interface  
4 component is to be executed at the client system.

1 21. The method of claim 19 wherein the dynamically generated protocol is an object  
2 protocol if the user-interface component is to be executed at the server system.

1 22. The method of claim 19 further comprising building an application program  
2 comprised of the new software components using an integrated development  
3 environment.

- 1    23. In a computer network, a computer system hosting an application program, the  
2        computer system comprising:  
3                a software component analyzer identifying a first one of the software  
4                components of the application program as a candidate for partitioning;  
5                a software component generator generating a plurality of new software  
6                components corresponding to the identified software component candidate and a  
7                protocol to be used by the new software components for communicating with  
8                each other over the network; and  
9                a transmitter transmitting one of the new software components to a client  
10          system over the network for execution at the client system and for  
11          communication with another of the new software components at the computer  
12          system using the generated protocol when the application program is executed.
- 1    24. The computer system of claim 23 wherein a predetermined criterion for  
2        identifying the first software component as a candidate for partitioning is that the  
3        software component includes a user-interface element.
- 1    25. The computer system of claim 23 wherein the first one of software components  
2        identified as a candidate is a JAVA Bean.
- 1    26. The computer system of claim 23 wherein one of the new software components  
2        generated at the server system wraps the first one of software components  
3        identified as a candidate for partitioning.

- 1       27. The computer system of claim 26 wherein the new software component that  
2                  wraps the identified software component candidate is the new software  
3                  component that is transmitted to the client system for execution.
- 1       28. The computer system of claim 27 wherein the generated protocol is a component  
2                  protocol.
- 1       29. The computer system of claim 26 wherein the new software component that  
2                  wraps the identified software component candidate remains at the server system  
3                  for execution at the server system.
- 1       30. The computer system of claim 29 wherein the generated protocol is an object  
2                  protocol.
- 1       31. The computer system of claim 23 wherein the identified software component  
2                  candidate includes a description that facilitates identification of the first one of the  
3                  software components as a candidate for partitioning.
- 1       32. The computer system of claim 23 wherein each generated new software  
2                  component corresponding to the identified software component candidate  
3                  includes a copy of the external interfaces of the identified software component  
4                  candidate.
- 1       33. An application builder tool for providing an integrated development environment  
2                  in which to construct an application program, the application builder tool  
3                  comprising:

4           a software component pallet listing a plurality of software components that are  
5   available for selection by an application program developer in constructing an  
6   application program; and

7           a software component splitter generating a plurality of new software components  
8   from one of the software components listed by the software component pallet, one of  
9   the new software components being generated for execution on a client system and  
10   another of the new software components being generated for execution on a server  
11   system, the splitter generating a protocol to be used by the new software components to  
12   communicate with each other.